Music Service

Description

This document provides an overview of the Music Service and the technical details on how the service can be used.

Music Service is a web service provided by a record company for convenient online creation of CDs (disks). Songs can be uploaded and included in the disks. Disks can be viewed simply as playlists containing multiple songs.

The service is mostly intended for use by musicians or their managers for uploading their music, creating virtual disks, and then sending the disks into production. The service may also be used by any music lovers for creating and maintaining playlists of their favorite songs, as well as listening to the playlists and songs.

SOAP API

Service name: MusicService

All type names, attributes and operations are named using camelCase naming style. If the name consist of only one word, it is all lowercase. If the name contains more than one word, the first word is lowercase and all following names with a first capital letter. There are no spaces allowed in the names.

Each request must have an attribute named `token` of string type attached to it which is used for authorization. Only requests with valid tokens will be properly processed.

Data manipulation requests are required to have a unique `requestCode` of integral type attached which prevents the request from being processed multiple times.

Below is the list of operations and their descriptions.

addSong

Upload a song to the service database.

Request `addSongRequest` details:

Name	Type	Comment
requestCode	integer	The unique code of the request. *Example: 193741*
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
name	string	The name of the new song. *Example: Sing Me To Sleep*
author	string	The author of the new song. *Example: Alan Walker*
lengthSeconds	integer	The length of the song in seconds. *Example: 183*

Example request:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:addSongRequest>
         <mus:requestCode>41</mus:requestCode>
         <mus:token>salajane</mus:token>
         <mus:name>MyFirstSong</mus:name>
         <mus:author>Ants</mus:author>
         <mus:lengthSeconds>300</mus:lengthSeconds>
      </mus:addSongRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `addSongResponse` details:

Name	Type	Comment
id	integer	The unique ID of the song. *Example: 31*
name	string	The name of the added song. *Example: Sing Me To Sleep*

Name	Type	Comment
author	string	The author of the added song. *Example: Alan Walker*
lengthSeconds	integer	The length of the added song in seconds. *Example: 183*

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <addSongResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <id>1</id>
         <name>MyFirstSong</name>
         <author>Ants</author>
         <lengthSeconds>300</lengthSeconds>
      </addSongResponse>
   </S:Body>
</S:Envelope>
```

getSong

Fetch the song from the service database with the given unique ID.

Request `getSongRequest` details:

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
id	integer	The unique ID of the song. *Example: 283*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:getSongRequest>
         <mus:token>salajane</mus:token>
         <mus:id>1</mus:id>
```

```
        </mus:getSongRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

Response `getSongResponse` details:

Name	Type	Comment
id	integer	The unique ID of the song. *Example: 31*
name	string	The name of the added song. *Example: Sing Me To Sleep*
author	string	The author of the added song. *Example: Alan Walker*
lengthSeconds	integer	The length of the added song in seconds. *Example: 183*

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <getSongResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <id>1</id>
         <name>MyFirstSong</name>
         <author>Ants</author>
         <lengthSeconds>300</lengthSeconds>
      </getSongResponse>
   </S:Body>
</S:Envelope>
```

getSongList

Fetch the list of all songs from the service database.

Request `getSongListRequest` details:

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:getSongListRequest>
         <mus:token>salajane</mus:token>
      </mus:getSongListRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `getSongListResponse` details:

Name	Type	Comment
song	XML object	Zero or more occurrences. Contains information about a song (see `getSongResponse`).

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <getSongListResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <song>
            <id>1</id>
            <name>MyFirstSong</name>
            <author>Ants</author>
            <lengthSeconds>300</lengthSeconds>
         </song>
      </getSongListResponse>
   </S:Body>
</S:Envelope>
```

addDisk

Add a new disk to the service database.

Request `addDiskRequest` details:

Name	Type	Comment
requestCode	integer	The unique code of the request. *Example: 193741*

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
name	string	The name of the disk. *Example: My First Disk*
author	string	The author of the disk. *Example: Myself*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:addDiskRequest>
         <mus:requestCode>2</mus:requestCode>
         <mus:token>salajane</mus:token>
         <mus:name>MyNewDisk</mus:name>
         <mus:author>Me</mus:author>
      </mus:addDiskRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `addDiskResponse` details:

Name	Type	Comment
id	integer	The unique ID of the added disk. *Example: 12314*
name	string	The name of the added disk. *Example: My New Disk*
author	string	The author of the added disk. *Example: Myself*
lengthSeconds	integer	The amount of seconds this disk would play for with all the songs accounted for. *Example: 4800*
diskSongList	XML object	Contains zero or more songs (see `getDiskSongListResponse`).

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <addDiskResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <id>0</id>
         <name>MyNewDisk</name>
         <author>Me</author>
         <lengthSeconds>0</lengthSeconds>
         <diskSongList/>
      </addDiskResponse>
   </S:Body>
</S:Envelope>
```

getDisk

Fetch the disk from the service database with the given unique ID.

Request `getDiskRequest` details:

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
id	integer	The unique ID of the disk. *Example: 872367*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:getDiskRequest>
         <mus:token>salajane</mus:token>
         <mus:id>0</mus:id>
      </mus:getDiskRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `getDiskResponse` details:

Name	Type	Comment
id	integer	The unique ID of the disk. *Example: 872367*

Name	Type	Comment
name	string	The name of the disk. *Example: My Favorite Songs*
author	string	The author of the disk. *Example: John Doe*
lengthSeconds	integer	The amount of seconds the disk would play for taking all songs into account. *Example: 56060*
diskSongList	XML object	A list of songs on the disk (see `getDiskSongListResponse`).

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <getDiskResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <id>0</id>
         <name>MyNewDisk</name>
         <author>Me</author>
         <lengthSeconds>300</lengthSeconds>
         <diskSongList>
            <diskSong>
               <song>
                  <id>1</id>
                  <name>MyFirstSong</name>
                  <author>Ants</author>
                  <lengthSeconds>300</lengthSeconds>
               </song>
               <startTimeSeconds>0</startTimeSeconds>
               <format>mp3</format>
            </diskSong>
         </diskSongList>
      </getDiskResponse>
   </S:Body>
</S:Envelope>
```

getDiskList

Fetch the list of all disks from the service database. There are four optional filters which can be attached to this request. Only the list of disks which pass each filter will be returned.

Request `getDiskListRequest` details:

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
minLengthSeconds	integer	Optional. The minimum duration of the disk in seconds. *Example: 600*
maxLengthSeconds	integer	Optional. The maximum duration of the disk in seconds. *Example: 3600*
minSongCount	integer	Optional. The minimum number of songs on the disk. *Example: 3*
maxSongCount	integer	Optional. The maximum number of songs on the disk. *Example: 20*

Example request (with no filters):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:getDiskListRequest>
         <mus:token>salajane</mus:token>
      </mus:getDiskListRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `getDiskListResponse` details:

Name	Type	Comment
disk	XML object	Zero or more occurrences. Contains information about the disk (see `getDiskResponse`).

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <getDiskListResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
```

```
         <disk>
            <id>0</id>
            <name>MyNewDisk</name>
            <author>Me</author>
            <lengthSeconds>300</lengthSeconds>
            <diskSongList>
               <diskSong>
                  <song>
                     <id>1</id>
                     <name>MyFirstSong</name>
                     <author>Ants</author>
                     <lengthSeconds>300</lengthSeconds>
                  </song>
                  <startTimeSeconds>0</startTimeSeconds>
                  <format>mp3</format>
               </diskSong>
            </diskSongList>
         </disk>
      </getDiskListResponse>
   </S:Body>
</S:Envelope>
```

addDiskSong

Add a song to a disk by providing the unique IDs of both entities. A relationship will be created between the two existing objects.

Request `addDiskSongRequest` details:

Name	Type	Comment
requestCode	integer	The unique code of the request. *Example: 193741*
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
diskId	integer	The unique ID of the disk to associate the song with. *Example: 3452*
songId	integer	The unique ID of the song to associate the disk with. *Example: 9623*
startTimeSeconds	integer	The number of seconds from the beginning of the disk until this song starts. *Example: 60*

Name	Type	Comment
format	string	The file format of the song on the disk. Must be one of “mp3”, “wav” or “ogg”. *Example: ogg*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:addDiskSongRequest>
         <mus:requestCode>432</mus:requestCode>
         <mus:token>salajane</mus:token>
         <mus:diskId>0</mus:diskId>
         <mus:songId>1</mus:songId>
         <mus:startTimeSeconds>0</mus:startTimeSeconds>
         <mus:format>mp3</mus:format>
      </mus:addDiskSongRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `addDiskSongResponse` details:

Name	Type	Comment
song	XML object	Contains information about a song (see `getSongResponse`).
startTimeSeconds	integer	The offset from the beginning of the disk until this song begins. *Example: 6782*
format	string	The file format of the song on the disk. Must be one of “mp3”, “wav” or “ogg”. *Example: mp3*

Example response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <addDiskSongResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <song>
            <id>1</id>
            <name>MyFirstSong</name>
            <author>Ants</author>
            <lengthSeconds>300</lengthSeconds>
         </song>
         <startTimeSeconds>0</startTimeSeconds>
         <format>mp3</format>
      </addDiskSongResponse>
   </S:Body>
</S:Envelope>
```

getDiskSongList

Get the list of all songs on a single disk by providing the unique ID of the disk.

Request `getDiskSongListRequest` details:

Name	Type	Comment
token	string	The API token. *Example: A9hL22e2RMb2Jhsk*
diskId	integer	The unique ID of the disk. *Example: 2763*

Example request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mus="http://www.ttu.ee/idu0075/2018/ws/164718/music">
   <soapenv:Header/>
   <soapenv:Body>
      <mus:getDiskSongListRequest>
         <mus:token>salajane</mus:token>
         <mus:diskId>0</mus:diskId>
      </mus:getDiskSongListRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response `getDiskSongListResponse` details:

Name	Type	Comment
diskSong	XML object	Zero or more occurrences. Contains information about the song file on the disk (see `addDiskSongResponse`).

Example response:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <getDiskSongListResponse xmlns="http://www.ttu.ee/idu0075/2018/ws/164718/music">
         <diskSong>
            <song>
               <id>1</id>
               <name>MyFirstSong</name>
               <author>Ants</author>
               <lengthSeconds>300</lengthSeconds>
            </song>
            <startTimeSeconds>0</startTimeSeconds>
            <format>mp3</format>
         </diskSong>
      </getDiskSongListResponse>
   </S:Body>
</S:Envelope>
```

REST API

The REST API documentation is in a separate file, generated by Swagger UI.